Denial of Service

Robert J. Fleischer

September 30, 2005

Taxonomy

Selective disenfranchisement; denial of service

Applicability

Any polling place organization that has a software-controlled "bottleneck".

Method

The perpetrator causes the "bottleneck" system (it could be a DRE or an optical scan machine) to slow down, stall, or crash. This could be achieved through any software intrusion or "hacking" technique. Long lines will form of waiting voters. If the waiting time gets too long (determined for each individual voter by their obligations, such as job or family, or in some cases the voter's physical condition) some percentage of voters will leave the queue without voting.

This bottleneck slowdown may be implemented selectively in precincts whose known demographics or politics favor the opponent of the perpetrator. Alternatively, the attacking software might simply observe the actual voting pattern and implement the slowdown only when the count so far favors the opponent.

Resource Requirements

The perpetrator must have the opportunity to introduce a software intrusion into the bottleneck system. Other than possibly observe the count to identify an opponent-favoring location, all the software need to be able to do is slow down or crash the machine.

The software intrusion can be introduced through communication lines, memory devices, or it can be embedded in the software as an act of sabotage any time before election day.

Potential Gain

It would take some simulation, based on a lot of assumptions, to quantify this. Waiting times of several to 10 hours were observed in the last general election.

Likelihood of Detection

People usually accept computer crashes or slowdowns. To most people, nothing will seem amiss when a computer slows or crashes. The intruding software can easily cover its tracks, and reloading the software will usually clear any traces. There were a lot of reports of crashes in the 2004 election -- nobody seems to think much about it. The count is not tampered with in this attack. Voters are merely deterred from voting. The counts are "correct".

Countermeasures

Preventative Measures

Physical and communication must be so good that software intrusions are impossible. Since software tampering can be introduced even at the factory, this level of security may be impossible.

Detection Measures

Detection is difficult -- when would crashes or slowdowns be obvious? In the 2004 election, there were precincts whose waiting times varied widely, from under an hour to many hours. Did anybody take that as proof of anything?

Techniques for detecting software changes, such as checksumming, can help detect changes introduced after the checksum was calculated.

Citations

(I need to find some references to the many reports of slow downs and crashes in 2004 and other elections.)

Retrospective

The "dark side" of "turning out the vote" activities is action taken to discourage votes for the opponent. One problem with electronic voting systems is that they can introduce new bottlenecks into the voting process at the polling place. DREs are especially bad in this respect. A single voter occupies DREs for an extended period of time while the voter reads the ballot and marks their choices. However, due to cost, DREs are typically in shorter supply than, say, booths for marking paper ballots. The occasional failure of a DRE, and the need to restart it (or simply take it out of service) aggravates this bottleneck and causes longer lines.